AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method to facilitate cache coherence with adaptive write updates, comprising:

initializing a cache to operate using a write-invalidate protocol;

monitoring a dynamic behavior of the cache during program execution, wherein monitoring the dynamic behavior of the cache involves maintaining a count for each cache line of the number of cache line invalidations the cache line has been subject to during program execution; and

if the dynamic behavior indicates that better performance can be achieved using a write-broadcast protocol, switching the cache to operate using the write-broadcast protocol.

2. (Original) The method of claim 1, wherein monitoring the dynamic behavior of the cache involves monitoring the dynamic behavior of the cache on a cache-line by cache-line basis.

3. (Original) The method of claim 2, wherein switching to the write-broadcast protocol involves switching to the write-broadcast protocol on a cache-line by cache-line basis.

4 (Canceled).

5. (Currently amended) The method of ~~claim 4~~ claim 1, wherein if the number of cache line invalidations indicates that a given cache line is updated frequently, switching the cache line to operate under the write-broadcast protocol.

6. (Original) The method of claim 5, wherein if a given cache line is using the write-broadcast protocol and the number of cache line updates indicates that the given cache line is not being contended for by multiple processors, switching the given cache line back to the write-invalidate protocol.

7. (Currently amended) The method of ~~claim 4~~ claim 1, wherein if a shared memory multiprocessor includes modules that are not able to switch to the write-broadcast protocol, the method further comprises locking the cache into the write-invalidate protocol.

8. (Original) The method of claim 1, wherein the write-invalidate protocol sends an invalidation message to other caches in a shared memory multiprocessor when a given cache line is updated in a local cache.

9. (Previously presented) The method of claim 1, wherein the write-broadcast protocol broadcasts an update to other caches in a shared memory multiprocessor when the given cache line is updated in a local cache.

10. (Currently amended) An apparatus to facilitate cache coherence with adaptive write updates, comprising:

an initializing mechanism configured to initialize a cache to a write-invalidate protocol;

an monitoring mechanism configured to monitor a dynamic behavior of the cache, wherein monitoring the dynamic behavior of the cache involves

maintaining a count of cache line invalidations initiated by each processor within a shared memory multiprocessor; and

a protocol switching mechanism configured to switch the cache to a write-broadcast protocol if the dynamic behavior indicates that better performance can be achieved using the write-broadcast protocol.

11. (Original) The apparatus of claim 10, wherein monitoring the dynamic behavior of the cache involves monitoring the dynamic behavior of the cache on a cache-line by cache-line basis

12. (Original) The apparatus of claim 11, wherein switching to the write-broadcast protocol involves switching to the write-broadcast protocol on a cache-line by cache-line basis.

13 (Canceled).

14. (Currently amended) The apparatus of ~~claim 13~~ claim 10, wherein if the count of cache line invalidations indicates that a given cache line is updated frequently in different caches of the shared memory multiprocessor, switching the cache to the write-broadcast protocol.

15. (Original) The apparatus of claim 14, wherein if the given cache line is using the write-broadcast protocol and the count of cache line invalidations indicates that the given cache line is being invalidated in only one cache, switching the cache to the write-invalidate protocol.

16. (Currently amended) The apparatus of ~~claim 13~~ claim 10, further comprising a locking mechanism configured to lock the cache into the write-

invalidate protocol if the shared memory multiprocessor includes modules that are not able to switch to the write-broadcast protocol.

17. (Original) The apparatus of claim 10, wherein the write-invalidate protocol involves sending an invalidate message to other caches within a shared memory multiprocessor when a given cache is written to.

18. (Original) The apparatus of claim 10, wherein the write-broadcast protocol involves broadcasting a data update message to other caches within a shared memory multiprocessor when a given cache is written to.

19. (Currently amended) A computing system that facilitates cache coherence with adaptive write updates, comprising:

a plurality of processors, wherein a processor within the plurality of processors includes a cache;

a shared memory;

a bus coupled between the plurality of processors and the shared memory, wherein the bus transports addresses and data between the shared memory and the plurality of processors

an initializing mechanism configured to initialize the cache to a write-invalidate protocol;

a monitoring mechanism configured to monitor a dynamic behavior of the cache, wherein monitoring the dynamic behavior of the cache involves maintaining a count of cache line invalidations initiated by each processor within a shared memory multiprocessor; and

a protocol switching mechanism configured to switch the cache to a write-broadcast protocol if the dynamic behavior indicates that better performance can be achieved using the write-broadcast protocol.

20. (Currently amended) A means to facilitate cache coherence with adaptive write updates, comprising:

an initializing means for initializing a cache to a write-invalidate protocol;

a monitoring means for monitoring a dynamic behavior of the cache, wherein monitoring the dynamic behavior of the cache involves monitoring the dynamic behavior of the cache on a cache-line by cache-line basis; and

a protocol switching means for switching the cache to a write-broadcast protocol if the dynamic behavior indicates that better performance can be achieved using the write-broadcast protocol.